FORCE-BASED REASONING FOR ASSEMBLY PLANNING AND SUBASSEMBLY STABILITY ANALYSIS

Sukhan Lee[1,2], Chunsik Yi[2] and Fu-Chung Wang[2]

Jet Propulsion laboratory[1]
California institute of Technology
Pasadena, CA 91109

Department of Computer Science[2]
University of Southern California
Los Angeles, CA 90089

Abstract

in this paper, we show that force-based reasoning, for identifying a cluster of parts that can be decomposed naturally by the applied force, plays an important role in selecting feasible subassemblies and analyzing subassembly stability in assembly planning. This is because force-based reasoning provides a new set of constraints which are distinctive from those from geometric reasoning. Such constraints not only enhance the planning efficiency but also allows stability analysis in terms of subassembly deformation due to the mating forces. Specifically, force-based reasoning can be applied during the subassembly identification to reduce the generation of a large number of candidate subassemblies, and applied after the subassembly identification to eliminate unstable subassemblies deformable during mating operation.

1 Introduction

Most of the current research in assembly planning focus on the pure geometric reasoning of part mating interferences using either forward or backward approaches [1-5]. However, the real assembly task depends also on some other practical factors such as interconnection forces and assembly line installation (e.g. assembly tools, fixtures). In order to bring assembly planning closer to the real world environment, these factors that affect the assembly planning in addition to the geometric constraints should also be taken into account. The force analysis of subassembly stability under the existence of gravity or external forces was addressed in [6], but the encapsulation of the force analysis into assembly planning seemed still missing.

Moreover, when a candidate subassembly is generated, it has to be tested for the stability conditions[7]. Several assembly planning systems have incorporated the stability check, but they have either very limited check such as a connection stability analysis[8], which tests if all parts in the subassembly are related to each other, and local freedom of motion analysis[9], which tests if a part is free to slide; or have a overly complex stability check to be practical[6].

The necessary stability conditions for subassemblies are that each subassembly is either self-stable or stable, with a holding devices, two mating subassemblies have a common stable assembly pose, and each of the mating subassemblies must maintain the geometric relationships of parts during the mating operations. Thus, the stability check needs both geometric and physical reasoning on subassemblies.

This paper presents a way to consider the interconnection force factors when generating an assembly plan. Such analysis not only affects the generation of assembly sequences but also provides a method for the stability analysis which consequently affects the assembly costs. A method is developed using the algorithm for solving maximal flow problem for such force analysis. This particular method is also used for more completed stability analysis of subassemblies.

2 Subassembly Identification with Force-Based Reasoning

The core of backward assembly planning is to find the valid feasible subassemblies which satisfy some necessary and sufficient constraint conditions such as accessibility, common local freedom of motion, free mating paths and so on. As stated previously, another important constraint is the interconnection force for assembly task. Without considering the real interconnection forces, an assembly planning system with only pure geometric reasoning may generate some infeasible assembly sequences. Usually, the interconnection force information is provided from the design specifications by the assembly designers. Basically, the interconnection forces are determined by many factors such as interconnection types, clearances, materials etc., which can be obtained from the design handbooks. In this paper, we assume that the force information is available from the design specifications. The detail of how to calculate the force is another issue which will not be addressed here.

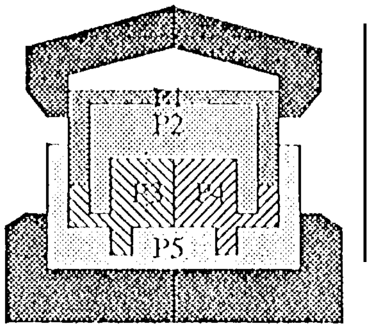
In the previous work [10], the interconnection feasibility of each liaison is tested by identifying the existence of paths that can propagate forces through them to the place where the connection or disconnection to be considered. The parts associated with a liaisons that violates the interconnection feasibility are then merged together. Because of considering individual liaison locally, this approach can only find the sufficient condition for the interconnection feasibility test. A new approach based on maximal flow and minimal cut-set method for such force analysis in assembly planning is developed. This approach is based on the analogy between the liaison graph of assembly and transport network. The detail is discussed in the following subsections.

2.1 Force Based Analysis Using Maximal flow Algorithm

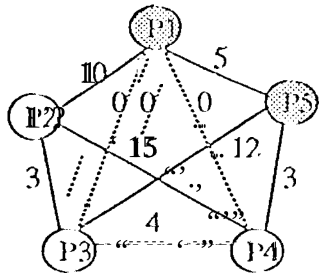
In the real assembly environment, the assembly task is usually constrained to operate in some certain directions due to the installation of assembly tools (such as robots), the fixtures and the design of the assembly as well. With such observation, instead of searching the whole cut-set space, possible subassemblies can be identified by considering only in those constrained directions with the assembly fixed in certain positions and orientations. Combining these environment constraints with the geometric and physical constraints, many undesired assembly sequences can be pruned out to increase the efficiency of assembly planning.

The maximal flow problem[12] is to find the maximal flow through a flow network with a source, sink and specified flow capacity of each link. The algorithm[13] used to solve this type of problem is adopted for the force analysis in assembly planning because the flow network model is similar to the liaison graph model used in assembly planning with the flow capacity of each link corresponding to the interconnection force of each liaison in liaison graph. The maximal flow that the network can carry is analogous to the minimal force that the tools can apply to decompose the assembly and the two subgraphs generated by the minimal cut-set found in the algorithm is analogous to the possible subassemblies generated under the force. The analogy between the flow network and liaison graph with force analysis is summarized as follows:

Flow Network	Liaison Graph
Capacity of link	Required Interconnection force of liaison
Flow	Force propagated through the liaisons
Maximal Flow	Minimal force to decompose the assembly
Minimal Cut-Sets	Possible direct subassemblies generated under the minimal force (maximal flow).



(a) Assembly on fixture



(b) Simplified Liaison Graph

Figure 1. A hypothetical assembly and its simplified liaison graph.

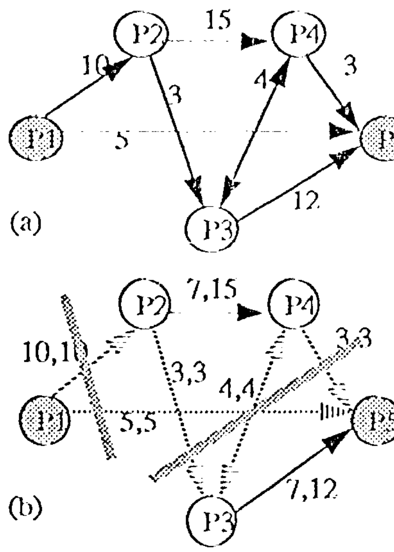


Figure 2. Flow network model of the assembly

The algorithms for solving the maximal flow problem had been proposed in many literatures [1 ?.-14], One of the algorithms which can find the maximal flow and identify one minimal cut set that is near the source is illustrated in Appendix A [13]. This algorithm is base.d on the algorithm developed by Ford and Fulkerson [14]. The time complexity of this algorithm is $O(V^2E)$ where V is the number of vertices and E is the number of edges in the network. 'l'he concept of how the maximal flow problem model is used for force. analysis of subassembly identification in backward assembly planning is illustrated in the following example.

Example: Maximal flow method for subassembly identification

A hypothetical assembly with P5 fixed in the assembly line, P1 as accessible part for the assembly tool and a particular assembly direction is shown in Figure. 1.(a). The simplified liaison graph of this assembly is shown in Figure. 1.(b). The number associated with each edge represents the interconnection force that is required to establish or brėak that liaison. The liaison graph of the assembly with the environment constraints is transformed into a flow network model as shown in Figure.2.(a). By using the algorithm listed in Appendix A, when the algorithm terminates, the maximal flow is found to be 15 and a minimal cut set nearest to the source is identified as S1={P1} and S2= (1'?, P3, 1'4, 1'5).

When the flow network reaches its maximal flow, those links that reach their full capacities will form a minimal cut-set (minimal cut-sets). The minimal cut-set ill flow network corresponds to the cut-set in the liaison graph of assembly which indicates the possible direct subassemblies under the interconnection force and environment constraints. In this example., two possible cut-sets of direct subassemblies are identified as $S^d_1|A$= {P1}, A-$S^d_1|A$= {1'?.,1'3,1'4,1'S) and $S^d_2|A$={P1,1'?.,1'4), A-$S^d_2|A$= {1'3,1'5) as shown in Figure.2.(b). By using the algorithm in Appendix A, the cut-set identified is the one that is near to the source. If there is no other disturbance involved, the cut-set that is nearest to the sour-cc will be the most likely direct subassembly that will be separated from the assembly because it is nearest to the assembly tool.

2.2 Complete Merging Process

The geometric reasoning for path existence checking is computationally expensive. In our previous work, a process called merging process was developed to identify those liaisons that should be merged together at current stage to avoid unnecessary path existence checking to increase the planning efficiency. However, in our previous approach, only the sufficient condition for merging process can be established due. 10 the locality of individual liaison, With this new method developed in this paper, the assembly is globally considered for the effects of interconnection forces. Therefore the sufficient and necessary conditions arc established for the merging process. While the flow network reaches its maximal flow, the pai 1s associated with the links whose actual flows arc still less than their capacities are merged together to avoid unnecessary tests. In general, the time complexity for the global path existence checking is $O(N^4)$. The worst case time complexity of the path existence checking for an assembly of N parts is $O(2^N N^4)$. With the. proposed merging process, the time complexity for path checking now becomes $O(k_1N^4 + V^2E)$ -- $O(k_2N^4)$ where k_1 and k_2 arc some constants. Thus the time needed for the path existence checking can be saved exponentially by paying only the polynomial time needed for the maximal flow algorithm. Consequently, This new complete merging process can improve the efficiency of assembly planning.

2.3 Subassembly Identification

Using the method discussed above, consideration of the interconnection force.s can be encapsulated into the backward assembly planning for the identification of feasible subassemblies. The algorithm for identifying the feasible subassemblies is summarized as follows:

Step 1): Identify the accessible parts for assembly tools. (Accessibility condition).
Step 2): Determine the holding parts and assembly dire.ctions.
Step 3): Transform the liaison graph into the flow network.
Step 4): Find the maximal flow and minimal cut-sets.
Step 5): Week the common local freedom of motion for the cut-sets in Step 4.
Step 6): Check the path existence for separating the cut-sets in step 4.
Step 7): Check the stability conditions for the subassemblies generated in step 6

To identify the true feasible subassemblies, the stability analysis is required for each subassemblies generated after step 6. The stability analysis is also needed for the analysis of assembly cost. Similar force based analysis discussed here may also be used for the stability analysis. The stability analysis with geometric and physical reasoning are discussed in following sections.

3 Geometric/force Reasoning in Stability Analysis

Any subassembl y formed during the process of decomposition, $S^f_i|A$, is subject to a stability condition test, Some unstable subassemblies arc not desirable because 1) the geometric relationships of parts composing the subassembly are not achievable (e.g., fig. 4.c), m 2) the physical force holding the parts relationships together cannot maintain the parts relationship.s during the mating operation. Smnc subassemblies may require assistance of holding devices in order to achieve the stability (e.g., fig. 4. b), but they may be more costly than the sc.lf-stable, subassemblies which do not require any extra holding devices (e.g., fig. 4a).

The process of identifying unstable subassemblies, thus, have two distinctive constraints that require geometric reasoning of subassembl y parts, and force-based reasoning of subassembly liaisons,

3.1 Geometric Reasoning in Stability Analysis

The stability of a subassembly can be defined in terms of a set of directions that a subassembly can separate freely. That is, a part or a group of parts together in a subassembly may be separated out in some direction without any resisting force except the gravitational force (frictional force is not considered). This set of directions is called Internal Freedom of Motion (IFM). This kind of stability may be analyzed strictly using geometric reasoning.

Let us define the following terms: A floating cluster, $F_k|(S^f_i|A)$, is a cluster of parts in $S^f_i|A$, and is connected to the rest of $S^f_i|A$ only by floating liaisons, liaisons with no physical force holding the parts together, and a disconnected cluster, $D_k|(S^f_i|A)$, is a cluster of par ts in $S^f_i|A$ that has no liaison connected to the rest of $S^f_i|A$. In an attributed liaison graph of $S^f_i|A$, $G_{LA}(S^f_i|A)$ where attributes such as mating-parts, mating-type, interconnection-type, and so on, arc associated with each liaison, $F_k|(S^f_i|A)$ corresponds to a connected subgraph of $G_{LA}(S^f_i|A)$ that can be separated from $G_{LA}(S^f_i|A)$ by a cut-set consisting only Of floating liaisons, called floating cut-set, and $D_k|(S^f_i|A)$ corresponds to a disconnected subgraph of $G_{LA}(S^f_i|A)$. The figure 3 illust rate.s disconnected clusters {P1,P2,P3} and (1'4), of the subassembly, $S^f_i|A$.

A floating cut-set, r_k, decomposes $S^f_i|A$ into $F_k|(S^f_i|A)$ and $S^f_i|A$-$F_k|(S^f_i|A)$. The cluster $S^f_i|A$-$F_k|(S^f_i|A)$, the complement cluster Of $F_k|(S^f_i|A)$ in $S^f_i|A$, must have an A-node, an accessible and manipulable node, of $S^f_i|A$ We can define the set of directions that the floating cluster, $F_k|(S^f_i|A)$, can be separated from $S^f_i|A$-$F_k|(S^f_i|A)$ by the local freedom of motion of $F_k|(S^f_i|A)$ against $S^f_i|A$-$F_k|(S^f_i|A)$, $LFM_k(S^f_i|A)$, defined as follows:

$$LFM_k(S^f_i|A) = \cap LFM(l_i; F_k|(S^f_i|A), S^f_i|A\text{-}F_k|(S^f_i|A))$$

fOr all l_i, l_i in r_k, and where $LFM(l_i)$, $l_i = (n_{i1}, n_{i2})$, is the freedom of motion of n_{i1} against n_{i2}. (Detailed analysis of freedom of motion can be found in [15])"

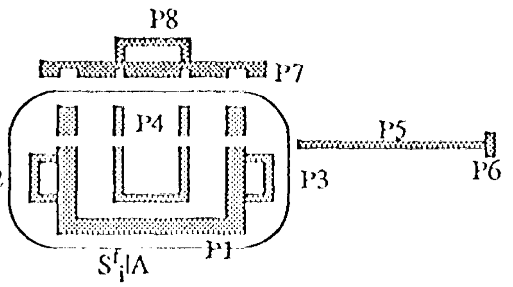
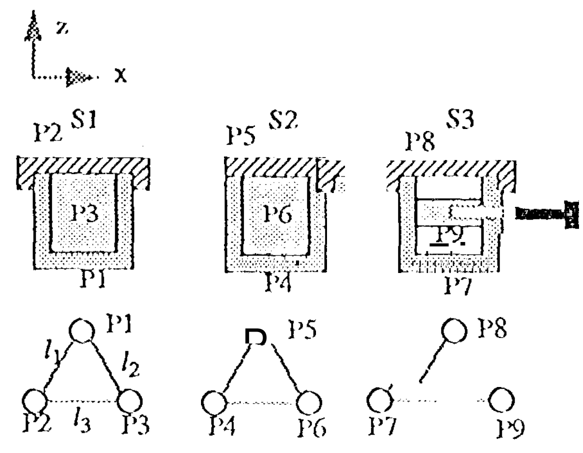


Figure 3: Disconnected Cluster



a) Self-stable b) Require Fixture c) Unstable

Figure 4: Subassembly Stability

The internal freedom of motion of $S^f_i|A$, $IFM(S^f_i|A)$ can be calculated by the following rules:

1) $IFM(S^f_i|A) = \emptyset$, if $\sim\exists D_k|(S^f_i|A)$ and $\sim\exists F_k|(S^f_i|A)$.
2) $IFM(S^f_i|A) = \cup LFM_k(r_k)$, $\forall r_k$, if $\sim\exists D_k|(S^f_i|A)$, but $\exists F_k|(S^f_i|A)$.
3) $IFM(S^f_i|A) = \{\pm x, \pm y, \pm z\}$, if $\exists D_k|(S^f_i|A)$.

Example: Internal Freedom of Motion

This example shows the process of calculating the internal freedom of motion of subassembly, S1, in the figure 4(a). The A-node of the subassembly S1 is the part PI, and all the liaisons arc floating liaisons. The liaisons arc defined as, l_1 = (P2,P1), l_2 = (P3,P1), and 13 = (P2,P3). The floating cut-sets are, $r_1 = \{l_1, l_2\}$, $r_2 = \{l_1, l_3\}$, and $r_3 = \{l_2, l_3\}$. The LFM(l_i), for i = 1, 2, 3 are, LFM(l_1) =- (-t z}, LFM(l_2) = {+z}, and LFM(l_3) = (-i z}. Note that, LFM(P2,P1) = -1 FM(P1,] '2.). Thus, -LFM(l_1) = {-z), -LFM(l_2) = {-z}, and -LFM(l_3) =- {-~,). The local freedom of motion of cut-sets are, LFM$_k$(r_1) = LFM(l_1) n LFM(l_2) = {+z}, LFM$_k$(r_2) = LFM(l_1) ∩ LFM(l_3) = {+z}, and LFM$_k$(r_3) = LFM(l_2) n - LFM(l_3) = { }, The internal freedom of motion of S1 is, IFM(S 1) = LFM$_k$(r_1) ∪ LFM$_k$(r_2) ∪ LFM$_k$(r_3) = {+z}.

Note that, the selection of LFM(l_i) or -LFM(l_i) must be consistent with the selected cut-sets. That is, l_i = (J '$_{i1}$,P$_{i2}$), l_i c r$_k$, such that all P$_{i1}$ must belong to the same cluster and all P$_{i2}$ must belong to the other cluster that are partitioned by the floating cut-set. That is why -LFM(l_3) was chosen fOr LFM$_k$(r_3): P1 and P2 belong to the same cluster. Based on the definition of IFM(S^f_i|A), we can establish the stability condition for S^f_i|A, as follows:

1) S^f_i|A is said to be self-stable, if IFM(S^f_i|A) is null, or IFM(S^f_i|A) has at most a single translational freedom of motion with a rotational freedom of motion only about the axis of translation. For example, the subassembly S 1 in the figure 4(a), with IFM(S1) = {+z}, is a self-stable subassembly.
2) S^f_i|A is said to be stable with the assistance of holding devices, if each D$_k$|(S^f_i|A) or F$_k$|(S^f_i|A), with more than a single translational freedom of motion, contains an A-Anode of the assembly *A*. This implies that the mating operation of S^f_i|A can be stabilized and completed with the assistance of external devices holding D$_k$|(S^f_i|A) or F$_k$|(S^f_i|A) Of more than a peg-and-hole type of motion freedom against S^f_i|A-F$_k$|(S^f_i|A). For example, the subassembly S2 in the figure 4(b), with IFM(S2) = {±x, +z}, requires a holding device.
3) Otherwise, S^f_i|A is said to be unstable. For example, the subassembly S3 in the figure 4(c), with IFM(S3) = {+z, -z}, is unstable because LFM$_k$({P9}, {P7,P8}) = LFM(P9,P7) n LFM(P9,P8) = {+z, -z}, where P9 is not an A-node.

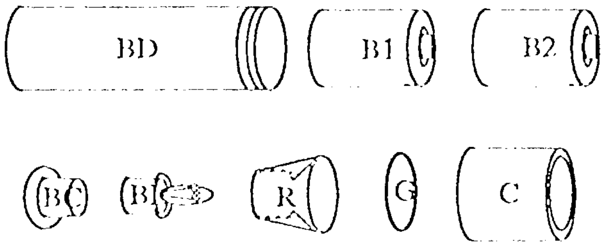
3, 1,1 Implementation of Internal Freedom of Motion

The efficiency of the algorithm for finding the floating cut-sets of a subassembly can be improved try using the contracting method. That is, the attributed liaison graph of subassembly can be reduced to a graph with lesser number of nodes, where each of new nodes is formed by collecting the parts connected by non-floating liaisons. A liaison is classified either floating or non-floating. Then, the algorithm for finding cut-sets, similar to [11], can be applied to this new graph. Our algorithm partitions the parts set into a set of all possible two components, and tests if each subgraph evoked by each part component forms a connected graph. A cut-set, a set of edges not included in either of the subgraphs, is a valid cut-set only if both subgraphs are connected graphs. For example, the flashlight assembly in the figure S shows the attributed liaison graph, the contracted graph, and the floating cut-sets. As it can be seen, the contracting method reduced the graph of 8 nodes to a new graph Of 4 nodes.

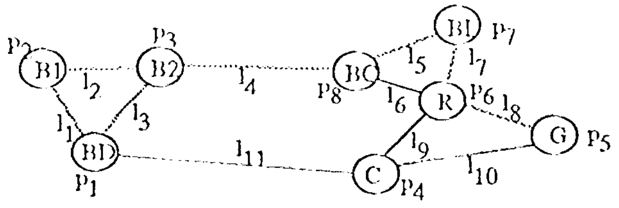
3., 1.2 Stable Common Assembly Pose

Even though S^f_i|A and A-S^f_i|A are stable with one or more stable assembly poses, they cannot be selected as feasible subassemblies if S^f_i|A and A-S^f_i|A do not have a common assembly pose. An assembly operation of S^f_i 1A and A-S^f_i|A is only possible if they have a common stable assembly pose.

Let us assume that the parts {P5, 1'6, P10} form a subassembly (figure 6) of an assembly, where the liaison l_4 has the mating type of insert and the interconnection type of attachment, and the liaison 1_5 has the mating type of insert and the interconnection type of press-fit. One possible cut-set for this subassembly is {l_5}. Then, the set of stable assembly poses for the subassembly {P6, 1'10] is {-z}, and for the subassembly {P5} is {±x,±y,+z} (figure 7.) Thus, the set of common stable assembly poses is {-z.],

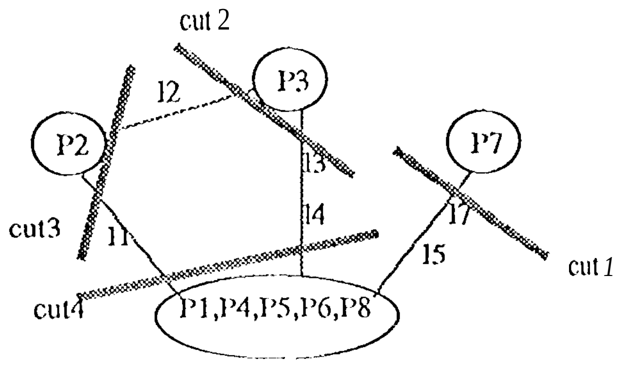


(a) An Exploded View



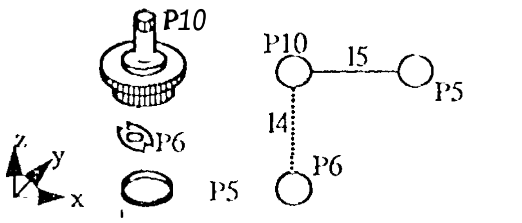
. . . . Float Liaisons ---- Nonfloating Liaisons

(b) Abstract Liaison Graph



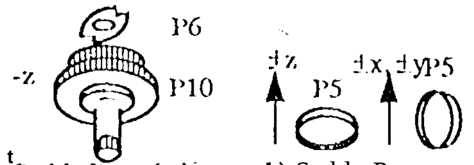
(c) Contracted Graph and Cut-Sets

Figure 5. Flash Light Assembl y



a) Gear Assembly b) Attributed Liaison Graph

Figure 6. Gear Subassembly



a) Stable Pose (-z) b) Stable Poses

Figure 7. Assembly Poses of Gear Subassemblies

3.2 Force-Based Reasoning in Stability Analysis

Geometric reasoning on a subassembly is not sufficient to test for the stability condition of feasible subassemblies. Some clusters of parts may form stable subassemblies, but they may not be stable to perform mating operations. The feasibility of applying force required for the interconnection of $S^f_i|A$ and $A\text{-}S^f_i|A$, while maintaining stability in each $S^f_i|A$ and $A\text{-}S^f_i|A$, requires reasoning on the force delivery to the mating liaisons through the intermediate liaisons.

To test the feasibility of applying force for the interconnection of mating liaisons, l_i, we may define the following: Internal Static Force Space of a subassem-My, $ISFS(d,S^f_i|A)$, analyzes the force deliverability of an external force applied to an A-node, through the intermediate liaisons, to the mating liaisons, l_i, in d direction, $d \in \{\pm x, \pm y, \pm z\}$. The complex network of intermediate liaisons must be analyzed such that each liaison does not receive more force than its form capacity (force required to separate. it in -d direction), and each mating liaison receives the amount of force equal to its mating force. in other words, $ISFS(d,S^f_i|A)$ is used to check if the geometric relationships among the parts can remain unchanged by the mist ing operation. We assume that parts arc rigid.

The force applied to a node. is equal to force out of a node, that is, $\sum F^{in}_i = \sum F^{out}_j$ (figure 8) where $i,j \geq 0$, and F^{in}_i is the form into a node and F^{out}_j is the force out of a node. Each liaison has its force capacity F^c, where $F^{in}_i < F^c_i$ and $F^{out}_j \leq F^c_j$.

Now, internal static fore.c space problem of $S^f_i|A$ may be restated in the following way. Find values of F^{in}_i and F^{out}_j for all liaisons in $S^f_i|A$ such that the following conditions hold: $\sum F^{in}_i = \sum F^{out}_j$, $F^{in}_i \leq F^c_i$, $F^{out}_j \leq F^c_j$, and $\sum F^{out}_k = \sum$ mating forces, for $\sum F^{out}_k$ of mating liaisons. The last condition tells us wllc.the.r the external force is deliverable to mating liaisons.

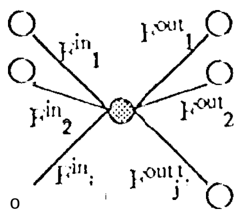


Figure 8. Force into a Node Equals to Force Out

$ISFS$ problem can be solved using a maximal-flow algorithm with some changes to the representation of $G_L(S^f_i|A)$, the attributed liaison graph. The maximal-flow algorithm is based on flow network model, FNM, that is defined as: A directed graph, with a source. node and a sink node; each edge is associated with nonnegative integer f(u,v) for all nodes other than the source or sink: $\sum f(u,v) = \sum f(v,u)$, where f(u,v) quantifies the flow from some u to some v; each edge is associated with c(u,v), called capacity: $0 \leq f(u,v) \leq c(u,v)$.

The result of applying the maximal-flow algorithm to FNM(d) is the maximal flow into the sink node from all mating node.s which is equivalent to maximal force (up to the force. required for the interconnection) that can be applied to the mating nodes from the A-node. Note that we must assign each mating liaison the force little larger than the actual force in FNM(d) because the external force must be greater than the mating force to complete the interconnection.

Now, we will describe the algorithm for $ISFS$ which have been implemented.

$ISFS(d,S^f_i|A)$ Algorithm:

```
Step 1) Transform $G_L(S^f_i|A)$ into FNM:
    - Create an array P of parts in $G_L(S^f_i|A)$ and add a
      special part *sink* to P.
    - For each liaison, (Pi,Pj), in $G_L(S^f_i|A)$ Do
        If the mating direction, c1 $\in$ LFM(Pi|Pj) then
            If FORCE(Pi|Pj) $\neq$ O then
                add the pair (Pj, FORCE(Pi|Pj)) to the Pi list
            Else
                add the pair (Pj, $\infty$) to the Pi list.
            If the mating direction, d $\in$ LFM(Pj|Pi) then
                If FORCE(Pj|Pi) $\neq$ O then
                add the pair (Pi, FORCE(Pj|Pi)) to the Pi list.
                add the pair (Pi, $\infty$) to the Pj list,
            (Note: FORCE(Pi|Pj) is the disassembly force
            of Pi from P2.)
    - For each mating parts, Pk $\in$ $G_L(S^f_i|A)$ Do
        Add the pair (*sink*,Fk) to the Pk list, where Fk
        is the mating force of Pk.
Step 2) Apply the Maximal-Flow-Algorithm
    to calculate the Maximal-Flow[12-14].
Step 3) If Maximal-blow equals to the sum of mating
    forces, then return STABLE,
    Otherwise, return UNSTABLE.
```

Example: Internal Static Force Space

The figure 9 illustrates *ISFS* analysis of two simple mating subassemblies (figure 9 (a)), with the mating direction equals to z. Both subassemblies consist of only three pai (s, and the corresponding attributed liaison graphs arc shown (figure 9 (b)). A-nodes of each subassemblies are colored dark. The subassemblies arc transformed into FNM as follows: FNM(-z) for the sub assembly consisting of parts{A,B,C}, and FNM(+z) for the subassembly consisting of parts{D,E,F}, where the capacity (force) of each edge (liaison) is shown on the right side of the paired value on each edge (figure 9 (c)). One of the possible solution of maximal flow is shown in the figui c 9 (c) (some symmetric directed edges are not di awn to make the figure clear, and they have flows equal to zero) for each subassemblies, The subassembly consisting of parts{A,B,C} has the sum of the maximal flow equal to the sum of the capacities to the sink, which means that it is stable for the mating operation, but the other subassembly has smaller sum of maximal flows than the sum of capacities, which means that it is unstable for the mating operation. Overall, this result indicates that the.se subassemblies cannot be chosen as feasible subassemblies.

We can now establish the stability condition of $S^f_i|A$ and $A\text{-}S^f_i|A$ in the following way:

1) Both $S^f_i|A$ and $A\text{-}S^f_i|A$ must be self-stable or stable with aid of holding devices.
2.) $S^f_i|A$ and $A\text{-}S^f_i|A$ must have a common stable assembly pose.
3) Internal static force space of both $S^f_i|A$ and $A\text{-}S^f_i|A$ must have the static force greater than the external force.

4 Conclusion

This paper contributes to automatic assembly planning with a distinctive constraint, say, interconnection forces, added to the assembly planning, and a new and more c. fficient methodology for decomposing a subassembly during the assembly planning. In addition, with the developed force-based reasoning method, more unstable subassemblies may be pruned out during earlier stages of subassembly decomposition phase. The force-based reasoning presented here is thus important and necessary to bring the. assembly planner closer to real world environment.

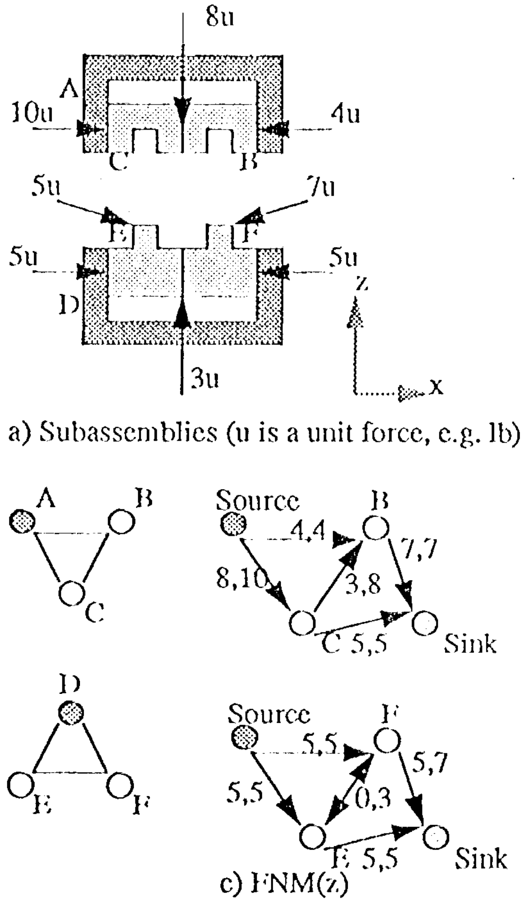


Figure 9. Internal Statice Force Space

Acknowledgments

This research described in this paper was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, and in part by the University of Southern California with the support from the National Science Foundation under grant CDR-87-17322 and from the industrial members of IMAR.

Appendix A: Algorithm for finding Maximal flow and Minimal cut-set [13]

```
Input: A network with source $a$, sink $z$, capacity $C$, and
vertices $a = v_0, \ldots, z = v_l$).
output: A maximal flow $F$ and a minimal cut-set $(P, \bar{P})$.
  1. [Initialize flow]:
     Set $F_{ij} = 0$ for each edge $(i,j)$
  2. [Label source]:
     Label vertex $a(\,, \infty)$
  3. [Sink labeled??]:
     If the sink z is labeled, go to step 6.
  4. [Next labeled vertex]:
     Choose the not yet examined, labeled vertex $v_i$
     with smallest index $i$.
     If none, stop (the flow is maximal;
          $P$ = set of labeled vertices, $\bar{P} = A - P$);
     else, set $v = v_i$
  5. [Label adjacent vertices]:
     Let $(\alpha, A)$ be the label of v. Examine each edge
     adjacent to v in the form of $(v,w), (w,v)$ [in the
     order $(v, v_0), (v_0,v), (v,v_1), (v_1,v), \ldots$]. where w is
     unlabeled.
     For an edge of the form $(v,w)$, if $F_{vw} < C_{vw}$,
          Label vertex w: $(v, \min\{A, C_{vw} - F_{vw}\})$,
          if $F_{vw} = C_{vw}$, do not label $w$;
     For an edge of the form $(w,v)$, if $F_{wv} > 0$,
          Label vertex w: $(v, \min\{A, F_{wv}\})$,
          if $F_{wv} = 0$, do not label w.
     Go to Step 3.
  6. [Revise the flow]:
     back track from z to find the path $p$ from $a$ to z.
     If the edge in $p$ is properly oriented,
          increase flow in c by A;
     otherwise decrease the flow by A.
     Go to step 2.
```

References

[1] R. J. Popplestone, A. P. Ambler and I. M. Bellos, "An Interpreter for a Language for Describing Assemblies.", *Artificial Intelligence Journal*, vol.14, pp. 79-107, 1980.

[2] S. Lee, "Disassembly Planning by Subassembly Extraction.", in *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems*, Aug., 1989, (Elsevier Science, MIT, MA), pp. 383-388.

[3] L. S. Homem de Mello and A. C. Sanderson, "AND/OR Graph Representation of Assembly Plans.", *IEEE Trans. Robotics & Automation*, vol. 6, pp 188-199, April 1990.

[4] T. L. De Fazio and D. E. Whitney, "Simplified Generation of All Mechanical Assembly Sequences.", *IEEE J. of Robotics & Automation*, pp. 640-658, Dec., 1987.

[5] S. Lee and Y. G. Shin, "Assembly Planning Based on Geometric Reasoning.", *Computers & Graphics, International Journal of Applications in Computer Graphics*, vol. 14, pp. 237-2s0, 1990,

[6] N. Boneschanscher, H. van der Drift, S. H. Backley, and R. H. Taylor, "Subassembly Stability," *Proceedings of AAAI, 1988, pp. 780-785.*

[7] L. S. Homem de Mello and A. C. Sanderson, "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences,' " *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 2,?,8-?40, April 1991.

[8] J. J. Waarts, N. Boneschanscher, and W. F. Bronsvoort, "A Semi-Automatic Assembly Sequence Planner," in *Proceedings of IEEE Conference on Robotics and Automation*, 1992, pp. 2?431-2438,

[9] R. H. Wilson and J. F. Rit, "Maintaining geometric dependencies in an assembly planner," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1990, pp. 890-895.

[10] S. Lee, "Backward Assembly Planning with Assembly Cost Analysis.", in *Proceedings of the 1992 IEEE Conference on Robotics & Automation*, 199?., pp. ?.38?.-?.391.

[11] L. S. Homem de Mello and A. C. Sanderson, "A Basic Algorithm for the Generation of Mechanical Assembly Sequences" in the book of *Computer-Aided Mechanical Assembly Planning*, chap 7, edited by L. S. Homem de Mello and S. Lee, Kluwer Academic Publishers, Boston, 1991.

[12] A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, 1985,

[13] R. Johnsonbaugh, *Discrete Mathematics*, 2nd ed., Macmillan Publishing Company, 1990.

[14] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, 196?.

[15] R. Wilson and J. Latombe, "On the Qualitative Structure of a Mechanical Assembly," in *AAAI*, 1992, pp. 697-70?,.